CO3 (Version 2.5)

Naoki Nishida and Misaki Kojima

Nagoya University, Nagoya, Japan
nishida@i.nagoya-u.ac.jp k-misaki@nagoya-u.jp

CO3, a **co**nverter for proving **co**nfluence of **co**nditional TRSs,[1] tries to prove confluence of conditional term rewrite systems (CTRSs, for short) by using a transformational approach (cf. [6]). The tool first transforms a given weakly-left-linear (WLL, for short) 3-DCTRS into an unconditional term rewrite system (TRS, for short) by using $\mathbb{U}_{conf}$ [2], a variant of the *unraveling* $\mathbb{U}$ [8], and then verifies confluence of the transformed TRS by using the following theorem: A 3-DCTRS $\mathcal{R}$ is confluent if $\mathcal{R}$ is WLL and $\mathbb{U}_{conf}(\mathcal{R})$ is confluent [1, 2]. The tool is very efficient because of very simple and lightweight functions to verify properties such as confluence and termination of TRSs.

Since version 2.0, a *narrowing-tree*-based approach [7, 3] to prove infeasibility of a condition w.r.t. a CTRS has been implemented [4]. The approach is applicable to *syntactically deterministic* CTRSs that are operationally terminating and *ultra-right-linear* w.r.t. the *optimized* unraveling. To prove infeasibility of a condition c, the tool first proves confluence, and then linearizes c if failed to prove confluence; then, the tool computes and simplifies a narrowing tree for c, and examines the emptiness of the narrowing tree. Since version 2.2, CO3 accepts both *join* and *semi-equational* CTRSs, and transforms them into equivalent DCTRSs to prove confluence or infeasibility [5].

This version has a new disproof criterion for confluence of CTRSs. A CTRS is trivially non-confluent if there exist a (possibly unconditional) critical pair $\langle s,t\rangle \Leftarrow c$ of $\mathcal{R}$ and a substitution θ such that θ satisfies c and $s\theta, t\theta$ are different constructor terms of $\mathcal{R}$. To find such a substitution θ, we implemented the following sufficient condition for $\langle s,t\rangle \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$: There exists some $i \in \{1,\ldots,k\}$ such that $s_i\theta \to^{\equiv}_{\mathcal{R}'} t_i\theta$, $s_i\theta, t_i\theta$ are constructor terms of $\mathcal{R}$, and for all $j \in \{1,\ldots,i-1,i+1,\ldots,n\}$, $s_j\theta \to_{\varepsilon,\mathcal{R}'} t_j\theta$, where $\mathcal{R}' = \{\ell \to r \mid \ell \to r \Leftarrow c \in \mathcal{R},\ c = \epsilon\}$. This criterion works for, e.g., `251.ari` and `319.ari` in ARI-COPS[2] (`262.trs` and `330.trs`, resp., in COPS[3]).

Example 1. *Consider the 3-DCTRS* `251.trs`*:*

$$\mathcal{R}_{251} = \left\{ \begin{array}{r} \mathsf{plus}(\mathsf{0},y) \to y \\ \mathsf{plus}(\mathsf{s}(x),y) \to \mathsf{plus}(x,\mathsf{s}(y)) \\ \mathsf{f}(x,y) \to z \Leftarrow \mathsf{plus}(x,y) \twoheadrightarrow \mathsf{plus}(z,z') \end{array} \right\}$$

The (conditional) critical pairs of $\mathcal{R}_{251}$ *are*

$$\langle z,w\rangle \Leftarrow \mathsf{plus}(x,y) \twoheadrightarrow \mathsf{plus}(z,z'),\ \mathsf{plus}(x,y) \twoheadrightarrow \mathsf{plus}(w,w')$$

The first condition $\mathsf{plus}(x,y) \twoheadrightarrow \mathsf{plus}(z,z')$ *is satisfied by the substitution* $\theta_1 = \{z \mapsto x, z' \mapsto y\}$*, which does not instantiate the second condition* $\mathsf{plus}(x,y) \twoheadrightarrow \mathsf{plus}(w,w')$*. The second condition is satisfied by the second rule of* $\mathcal{R}_{251}$ *by means of the substitution* $\theta_2 = \{x \mapsto \mathsf{s}(x'), w \mapsto x', w' \mapsto y\}$*:* $\mathsf{plus}(x,y)\theta_2 \to_{\mathcal{R}_{251}} \mathsf{plus}(w,w')\theta_2$*. The composed substitution* $\theta = \{z \mapsto \mathsf{s}(x'), z' \mapsto y, x \mapsto \mathsf{s}(x'), w \mapsto x', w' \mapsto y\}$ *is a constructor substitution satisfying the conditional part of the critical pair. For the composed substitution* θ*,* $z\theta$ *and* $w\theta$ *are not joinable because they are different constructor terms. Therefore,* θ *is a witness disproving confluence of* $\mathcal{R}_{251}$*.*

[1] http://www.trs.css.i.nagoya-u.ac.jp/co3/
[2] https://ari-cops.uibk.ac.at/ARI/
[3] https://ari-cops.uibk.ac.at/COPS/

References

[1] K. Gmeiner, B. Gramlich, and F. Schernhammer. On soundness conditions for unraveling deterministic conditional rewrite systems. In A. Tiwari, editor, *Proceedings of the 23rd International Conference on Rewriting Techniques and Applications*, volume 15 of *Leibniz International Proceedings in Informatics*, pages 193–208, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012.

[2] K. Gmeiner, N. Nishida, and B. Gramlich. Proving confluence of conditional term rewriting systems via unravelings. In N. Hirokawa and V. van Oostrom, editors, *Proceedings of the 2nd International Workshop on Confluence*, pages 35–39, 2013.

[3] Y. Maeda, N. Nishida, M. Sakai, and T. Kobayashi. Extending narrowing trees to basic narrowing in term rewriting. IEICE Technical Report SS2018-39, Vol. 118, No. 385, pp. 73–78, the Institute of Electronics, Information and Communication Engineers, 2019 (in Japanese).

[4] N. Nishida. CO3 (Version 2.1). In M. Ayala-Rinćon and S. Mimram, editors, *Proceedings of the 9th International Workshop on Confluence*, page 67, 2020.

[5] N. Nishida. CO3 (Version 2.2). In S. Mimram and C. Rocha, editors, *Proceedings of the 10th International Workshop on Confluence*, page 61, 2021.

[6] N. Nishida, T. Kuroda, and K. Gmeiner. CO3 (Version 1.3). In B. Accattoli and A. Tiwari, editors, *Proceedings of the 5th International Workshop on Confluence*, page 74, 2016.

[7] N. Nishida and Y. Maeda. Narrowing trees for syntactically deterministic conditional term rewriting systems. In H. Kirchner, editor, *Proceedings of the 3rd International Conference on Formal Structures for Computation and Deduction*, volume 108 of *Leibniz International Proceedings in Informatics*, pages 26:1–26:20, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

[8] E. Ohlebusch. Termination of logic programs: Transformational methods revisited. *Applicable Algebra in Engineering, Communication and Computing*, 12(1/2):73–116, 2001.